

Dan Mopsick

Dr. Juan Arias

Software Development 1

April 7 2017

CCD Parser Project Milestone

Abstract

The goal of this project is to create a system that is capable of parsing XML Continuity of Care into Java objects and storing these objects in a database that can be accessed with a RESTful API. In order to parse a file, the filename must be passed to the system. Currently the system can handle hard coded in files. Upon final release it will be able to take in a file name from the user and store it into the database. The system also has the capability to display specific patients or all patients that are saved in the database.

Introduction

The motivation of this work is to create a easy to use and scalable system for transferring patient information from healthcare providers to the patients' themselves. The major issue of transferring patient information is patient confidentiality and security. This is an issue that extends farther than the scope of the project. The purpose of this project is to create the infrastructure for the passing of patient information and the establishing of protocols to process and read patient information from XML into Java. The RESTful aspects of the system are built using [Spring Data and Spring Boot](#). The XML files are parsed utilizing javax's parsing package.

Detailed System Description

The system in its current state can only be used by the user to receive information from a premade example file. In the final version the user will be able to parse any XML file that they put in the root directory given that the file follows the protocol laid out in the sample file. This will be done by having the user use [PostMan](#) (a service that allows the user to reach the resting endpoints of the RESTful web service created in this project) to make a POST request that passes the name of the file that the user wishes to parse and add to the database. In its current state a user can access a specific patient by making a GET request with the id of the desired patient in the path. (ex: <http://localhost:8080/patient/1>). A user can return all of the patients saved into the database by making an empty GET request (<http://localhost:8080/patient/>). The system is started by `Application.java`. The class `PatientController` handles all of the requests from the user and uses `PatientService` and `ParserService` to interact with the `PatientRepository` and `XMLDomParser` respectively. This allows for loose coupling between the classes and classes that are built with one purpose in mind.

Requirements

The specific problem that this system is addressing is the lack of digital availability of a patient's medical records. Individual practices and healthcare providers currently have their own systems, but there is no standard system capable of efficiently handling patient information. This project seeks to build the infrastructure capable of storing patient information in a database and providing access to this database. The project seeks to build the infrastructure in such a way that it would be scalable to multiple healthcare providers and large amounts of patients. This project seeks to do this by creating a simple and consistent RESTful API.

Literature Survey

Continuity of Care Documents were developed to provide a protocol for storing patient information in a standardized format within XML file. This is a step in the right direction. The XML format used in this project is based directly on the CCD format. For ease of access and in order to focus on the programming of this project, a simplified format was utilized. This project seeks to extend the protocol of CCDs to functional databases and RESTful web services that will provide patients and healthcare providers with access to this standardized information.

User Manual

This project is purely a backend project, which means that in its current state there is no implementation of the front end. A user will access this system through Postman as mentioned above. In this current state of the project, the only commands that are supported are two GET commands. The user can access a specific patient's information by making a call with the desired patient's id. The user can see the information of all the patients in the database by making a GET call with no path variable. In the final implementation a user will be able to manually had a file to the project directory and add it to the database by passing the name of the file as a path variable via a POST command (<http://localhost:8080/patient/fileName>).

Conclusion

In conclusion, this project accomplishes it's goal of creating a basic infrastructure for facilitating the transfer of patient information. To be a more useable system, the ability for the user to add patients into the database from files must be added. The issues of scalability could be lessened by the addition of search functions. Searching by id becomes increasingly hard as the number of patients increases. This project does create a RESTful web services that provides a simple way to view and store patient information from healthcare providers.

Reference/ Bibliography

Freeman, George, et al. "Continuity of care." Report of a scoping exercise for the National Co-ordinating Centre for NHS Service Delivery and Organisation R & D (NCCSDO). London: NCCSDO (2001).

Walker, Jan, et al. "The value of health care information exchange and interoperability." Health affairs 24 (2005): W5.